

**ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПЛКИ
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»**

**ІНСТИТУТ ЕКОНОМІКИ, УПРАВЛІННЯ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
ФОРМА НАВЧАННЯ ДЕННА
КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ТА СОЦІАЛЬНОЇ
ІНФОРМАТИКИ**

Допускається до захисту

Завідувач кафедри _____ О.О. Ємець
(підпис)

«_____» _____ 2020 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА ДО ДИПЛОМНОЇ РОБОТИ

на тему

**Розробка тренажеру з теми «Метод квазімінорів для відшукування
найдовшого та найкоротшого шляхів» дисципліни «Теорія систем
та математичне моделювання»**

зі спеціальності 122 «Комп'ютерні науки»

Виконавець роботи Заможський Олександр Олександрович

_____ «___» _____ 2020р.
(підпис)

Науковий керівник к.ф.-м.н., доц. Олексійчук Юрій Федорович

_____ «___» _____ 2020р.
(підпис)

ПОЛТАВА 2020р

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	3
ВСТУП	4
1.ПОСТАНОВКА ЗАДАЧІ	6
2.ІНФОРМАЦІЙНИЙ ОГЛЯД	8
2.1. Задача про найкоротший шлях	8
2.2. Аспекти роботи із алгоритмами	9
2.3 Актуальність теми роботи	9
3.ТЕОРИТИЧНА ЧАСТИНА	11
3.1. Робота з квазімінорами	11
3.2 Алгоритм тренажеру	11
3.3. Блок-схема алгоритму	16
4.ПРАКТИЧНА ЧАСТИНА	26
4.1 Інструкція по роботі з програмою	26
4.2. Опис програми	54
ВИСНОВКИ	61
СПИСОК ЛІТЕРАТУРНИХ ДЖЕРЕЛ	62
ДОДАТОК А.	63

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Умовні позначення, символи, одиниці, скорочення, терміни	Пояснення умовних позначень, символів, одиниць, скорочень, термінів
Граф	Це сукупність об'єктів із зв'язками між ними. Об'єкти розглядаються як вершини, або вузли графу, а зв'язки — як дуги, або ребра.
Квазімінор	Квазімінором M_{lk} елемента a_{lk} матриці A називають матрицю, яку одержують, викресливши в A рядок l та стовбець k .
C#	Об'єктно-орієнтована мова програмування

ВСТУП

Сучасне життя все частіше потребує автоматизації процесів мислення та компактної подачі інформації. У людей все частіше не вистачає часу для поглиблення у якусь сферу розуміння, тому людство шукає методи для збільшення числа операцій, які можна здійснювати, не роздумуючи над ними. Відшукування найшвидшого та найякіснішого методу для досягнення своєї мети, одна з найважливіших задач нашого часу. З точки зору програмування пошуку таких шляхів, приділяється значна увага.

Отже, розробка тренажеру на дану тему, для поглиблення знань з пошуку шляхів є **актуальною та важливою задачею!**

Метою бакалаврського проекту є розробка тренажеру з теми «Метод квазімініорів для відшукування найдовшого та найкоротшого шляхів» для дистанційного начального курсу «Теорія систем та математичне моделювання».

Задачі дипломного проектування:

- 1) Дослідити тему «Метод квазімініорів»;
- 2) Підібрати декілька прикладів матриць або графів для розробки тренажеру;
- 3) Розробити алгоритм тренажеру;
- 4) Створити блок-схему алгоритму тренажеру;
- 5) Розробити програму;
- 6) Описати створену програму;
- 7) Передати програму на впровадження.

Об'єкт дослідження – квазімініор, метод квазімініорів, відшукування шляхів.

Предмет дослідження – Метод квазімініорів.

При реалізації бакалаврської роботи, було використано середовище розробки Visual Studio. При написанні програми використано мову C#.

Особистий вклад – Самостійно створені блок-схема алгоритму, програма.

Практична новизна роботи – Створено перший тренажер з теми «Метод квазімінімумів для відшукування найдовшого та найкоротшого шляхів»

Програма повністю готова для використання.

Бакалаврська робота складається з чотирьох частин – постановка задачі, інформаційний огляд тренажеру, теоретичної та практичної частин. В першому розділі розглянуто постановку задачі. У другому надано означення використаного методу. В третьому розглянуто алгоритм та блок-схему. В четвертому розділі вказано як працює програма, та опис її функціоналу.

Обсяг пояснювальної записки: 35 стор., в т.ч. основна частина 9 стор., додатки - 17 стор., джерел – 6 назв.

1. ПОСТАНОВКА ЗАДАЧІ

Завдання дипломного проекту зводиться до того, щоб створити комп'ютерний тренажер з теми «Метод квазімінорів для відшукування найдовшого та найкоротшого шляхів» для дистанційної дисципліни «Теорія систем та математичне моделювання».

Для виконання завдання, подані нижче алгоритми (приклади, задачі).

В тренажері використано 2 приклади, для погляду на метод під різним кутом.

Приклад 1.

Дано матрицю 6х6.

	1	2	3	4	5	6	
		1					1
			1	1		1	2
				1	1		3
A=						1	4
				1		1	5
							6

де

$$d_{ij} = \begin{cases} a_{ij}, & a_{ij} \neq 0; \\ 0, & i = j; \\ \infty, & a_{ij} = 0, i \neq j, \end{cases}$$

для знаходження найкоротшого шляху та $\|D_{ij}\|$, де

$$D_{ij} = \begin{cases} -a_{ij}, & a_{ij} \neq 0; \\ 0, & i = j; \\ \infty, & a_{ij} = 0, i \neq j, \end{cases}$$

для знаходження найдовшого шляху.

Знайти найдовший та найкоротший шляхи з точки 1 в точку 6.

Приклад 2.

Дано граф з п'ятьма вершинами (рис. 1.1).

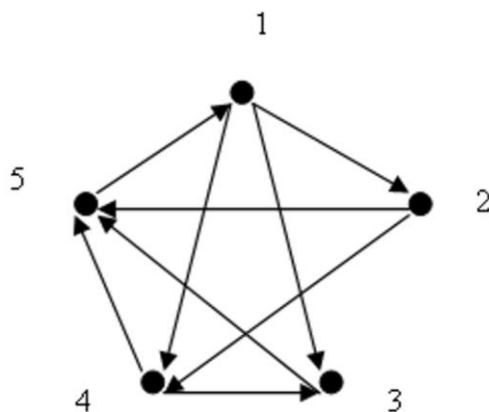


Рисунок 1.1 – Граф

Знайти найдовший та найкоротший шлях з вершини 1 у вершину 5.

Розробити тренажер, який реалізує алгоритм проходження даних завдань.

Перед проходженням тренажеру, слід ознайомитися з лекціями з поточної дисципліни «Теорія систем та математичне моделювання».

Перевірити програму на працездатність, протестувати.

Створити документацію по використанню програми.

2.ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Задача про найкоротший шлях.

В теорії графів, **задача про найкоротший шлях** полягає в знаходженні такого шляху між двома вершинами (або вузлами) графу, що сума ваг ребер з яких він складається мінімальна. Прикладом може бути знаходження найкоротшого шляху між двома пунктами на дорожній мапі; в цьому випадку, вершинами є пункти, а ребрами — відтинки дороги із вагами, рівними часу, необхідному для подолання цього відрізка.

Існує багато методів для розв'язання цієї задачі:

Алгоритм Дейкстри

Цей алгоритм названий за іменем творця і був розроблений 1959 р. У процесі виконання алгоритм перевірить кожен з вершин графа, і знайде найкоротший шлях до початкової вершини. Розв'язує задачі з однією парою, одним входом і одним виходом.

Алгоритм Беллмана-Форда

Алгоритм пошуку найкоротшого шляху у графі. Алгоритм знаходить найкоротший шлях з одної вершини до усіх інших. На відміну від алгоритму Дейкстри, алгоритм Беллмана – Форда допускає ребра з від'ємними вагами.

Алгоритм пошуку A* (A «із зірочкою»)

Уперше описаний 1968 р. Пітером Хартом, Нільсом Нільсоном і Бертрамом Рафаелем. Розв'язує задачу для однієї пари із використанням евристики в спробі пришвидшити пошук.

Алгоритм Флойда-Воршелла

Динамічний алгоритм для знаходження найкоротших відстаней між усіма вершинами орієнтованого графа. Розроблений у 1962 році Робертом Флойдом та Стівеном Воршеллом. Алгоритм уперше розробив та опублікував Бернард Рой.

Алгоритм Джонсона

Дозволяє знайти найкоротші шляхи між усіма парами вершин зваженого орієнтованого графа. Розв'язує задачу для всіх пар, і може бути швидшим за алгоритм Флойда-Воршелла на розріджених графах.

Інформацію що до використання цих методів, не важко знайти, а ось стосовно методу за темою дипломної роботи, вже на багато важче.

Метод, запропонований Г.Н. Поваровим дещо відрізняється від розглянутих вище методів і полягає у визначенні всіх шляхів між заданими вершинами і в подальшому розподілі їх по довжині. Для визначення шляхів між заданою парою вершин у графі, Г.Н. Поваров пропонує визначники особливого виду – **квазімінори**.

2.2. Аспекти роботи із алгоритмами

До позитивних аспектів можна віднести:

- Чіткість та зрозумілість алгоритмів;
- Різноманітність алгоритмів;
- Практична значність алгоритмів;

До негативних аспектів можна віднести:

- Відсутність тренажерів для деяких алгоритмів;
- Подання матеріалу;
- Робота деяких алгоритмів займає дуже багато часу;
- Складність вибору алгоритмів для застосування;

2.3. Актуальність теми роботи

Специфіка навчального процесу на підготовчому факультеті, все більше і більше змушує відводити особливе місце самостійній роботі студентів. Водночас, загальну інформацію завжди можна знайти в Інтернеті або в відповідній документації, а ось інформацію щодо ефективного її використання

знайти складніше, а всі наявні відомості досить фрагментарні, та їх важко зводити в єдине ціле. Для полегшення освоєння широкого спектру інформації, створюються спеціальні тематичні програми – тренажери.

Особливістю роботи тренажерів, є зібрана, та оброблена інформація, подана у релевантній формі, яка полегшує та прискорює сприйняття інформації студентом та подана у візуалізованій формі, що дає змогу, користувачу бачити результат своєї роботи.

Оскільки програм на тему «Метод квазімінорів для відшукування найдовшого та найкоротшого шляхів» не було виявлено, всі елементи даного тренажеру розроблялися лише на підставі теоретичних знань з даної теми без можливості використання практичних матеріалів з розробки саме такої програми. Тобто, ця програма є фактично унікальною авторською розробкою, яка відображає особисте бачення автору стосовно тренажеру, який потрібен для поглиблення розуміння даного методу іншими студентами.

3. ТЕОРИТИЧНА ЧАСТИНА

3.1. Робота з квазімінорами

Квазімінором M_{lk} елемента a_{lk} матриці A називають матрицю, яку одержують, викресливши в A рядок l та стовбець k . Позначимо $M_{lk} = |a_{ij \rightarrow lk}|_{kl}$

Будуємо матрицю безпосередніх шляхів $A = \|a_{ij}\|$, тобто таку матрицю A , де $a_{ij} \neq 0$ показує, що є шлях з i в j , $a_{ij} = 0$ – в іншому разі.

Викреслюємо в вихідній матриці стовбець з номером k , що відповідає номеру вершини, з якої починається шлях, та рядок з номером l , що відповідає номеру вершини, в кій цей шлях закінчується: $\pi_{kl} = |a_{ij \rightarrow lk}|_{kl} = M_{lk}$

При $k \neq l$ шляхи з k в l символічно представляються як

$$\pi_{ke} = M_{lk} = |a_{ij \rightarrow lk}|_{kl} = \sum_{\forall j \neq k} a_{kj} A_{kj} \quad (1)$$

Де A_{kj} – квазімінор, що містить на один рядок та на один стовпець менше, ніж M_{lk} , та $A_{kj} = 1$, при $j = l$ (тобто коли він відповідає елементу, що стоїть в стовпці з номером первісно викресленого рядка).

Якщо в (1) маємо доданок $a_{ij} A_{ij}$, то квазімінор A_{ij} далі розкладається по рядку, що мав в вихідній матриці номер j .

Кількість шляхів a_{kl} з вершини k в l одержимо, поклавши в π_{kl} всі $a_{ij} = 1$.

3.2 Алгоритм тренажеру

Тренажер містить декілька прикладів. Один з прикладів для тренажеру було взято з теми «Метод квазімінорів для відшукування найдовшого та найкоротшого шляхів».

Кожне питання у випадку неправильної відповіді виводить користувачу повідомлення з підказкою. У випадку правильності відповіді відбувається перехід до наступного кроку.

Для кожного прикладу умова видима під час усіх питань.

Розглянемо алгоритм для перших двох прикладів.

Приклад 1.

Дано матрицю 6х6. Знайти найкоротший шлях з вершини 1 у вершину 6.

Крок 1.

Дано матрицю: 1* 2 3 4 5 6

$$A =$$

0	1	0	0	0	0	1
0	0	1	1	0	1	2
0	0	0	1	1	0	3
0	0	0	0	0	1	4
0	0	0	1	0	1	5
0	0	0	0	0	0	6*

Який рядок та стовбець потрібно викреслити, аби розпочати?

Правильна відповідь 6 рядок 1 стовбець.

Крок 2.

Якою буде нова матриця?

Правильна відповідь: 2 3 4 5 6*

$$\delta_{16} =$$

1	0	0	0	0	1*
0	1	1	0	1	2
0	0	1	1	0	3
0	0	0	0	1	4
0	0	1	0	1	5

Крок 3.

В якій точці ми знаходимось, та в яку точку нам потрібно рухатися далі?

Правильна відповідь: Знаходимось у точці 2, потрібно рухатися у точку 6
= 2,6.

Крок 4.

Яка довжина шляху, який ми пройшли, та який саме це шлях?

Правильна відповідь: Довжина шляху = 2; Шлях = 1-2-6.

Приклад 2.

Дано матрицю 6х6. Знайти найдовший шлях з вершини 1 у вершину 6.

Крок 1.

Дано матрицю: 1* 2 3 4 5 6

$$A =$$

0	1	0	0	0	0	1
0	0	1	1	0	1	2
0	0	0	1	1	0	3
0	0	0	0	0	1	4
0	0	0	1	0	1	5
0	0	0	0	0	0	6*

Що потрібно зробити, аби розпочати?

Правильна відповідь: Викреслити 6 рядок та 1 стовбець.

Крок 2.

Дано Матрицю: 2 3 4 5 6*

$$\delta_{16} =$$

1	0	0	0	0	1*
0	1	1	0	1	2
0	0	1	1	0	3
0	0	0	0	1	4
0	0	1	0	1	5

Які рядок та стовбець, ми викреслюємо цього разу?

Правильна відповідь: 1 рядок та 1 стовбець.

Крок 3.

Якою буде матриця на цьому ході?

Правильна відповідь:

$$\delta_{16} = 1 * \begin{array}{c|c|c|c} & 3 & 4 & 5 & 6^* \\ \hline 2^* & 1 & 1 & 0 & 1 \\ 3 & 0 & 1 & 1 & 0 \\ 4 & 0 & 0 & 0 & 1 \\ 5 & 0 & 1 & 0 & 1 \end{array}$$

Крок 4.

Заповніть перший рядок отриманих матриць.

Правильна відповідь:

$$\delta_{16} = 1 * 1 + \begin{array}{c|c|c} & 4 & 5 & 6^* \\ \hline 3^* & 1 & 1 & 0 \\ 4 & 0 & 0 & 1 \\ 5 & 1 & 0 & 1 \end{array}$$

$$\delta_{16} = 1 * 1 + \begin{array}{c|c|c} & 3 & 5 & 6^* \\ \hline 3 & 1 & 1 & 0 \\ 4^* & 0 & 0 & 1 \\ 5 & 1 & 0 & 1 \end{array}$$

Крок 5.

$$\delta_{16} = 1 * 1 + \begin{array}{c|c|c} & 4 & 5 & 6^* \\ \hline 3^* & 1 & 1 & 0 \\ 4 & 0 & 0 & 1 \\ 5 & 1 & 0 & 1 \end{array}$$

Оберіть який з остаточних шляхів, буде найдовшим?

Правильна відповідь: 3-5-4-6.

3.3. Блок-схема алгоритму

На рис. 3.1-3.9 зображена блок-схема алгоритму(першого блоку завдань).

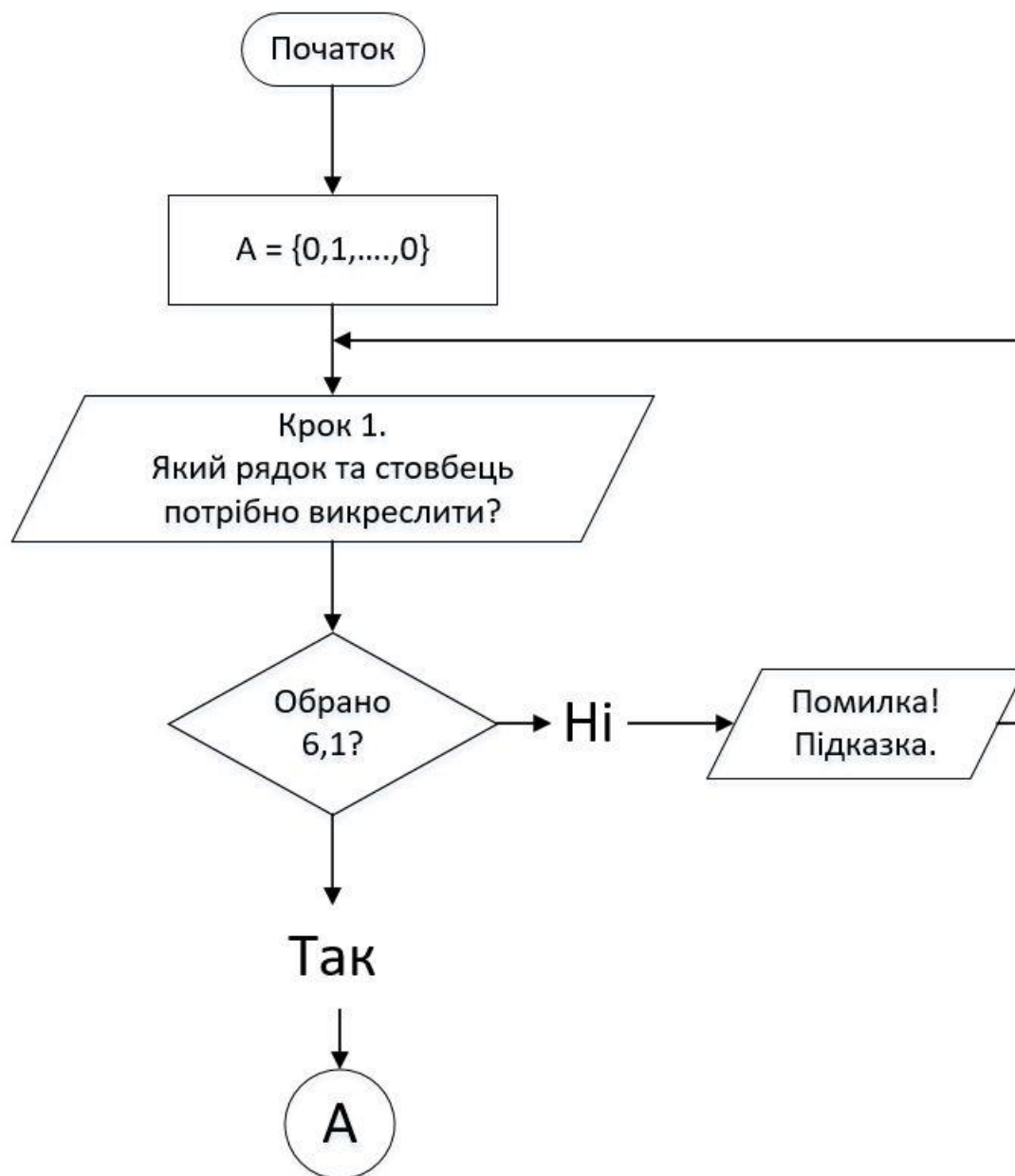


Рисунок 3.1 – Блок-схема алгоритму (тест 1)

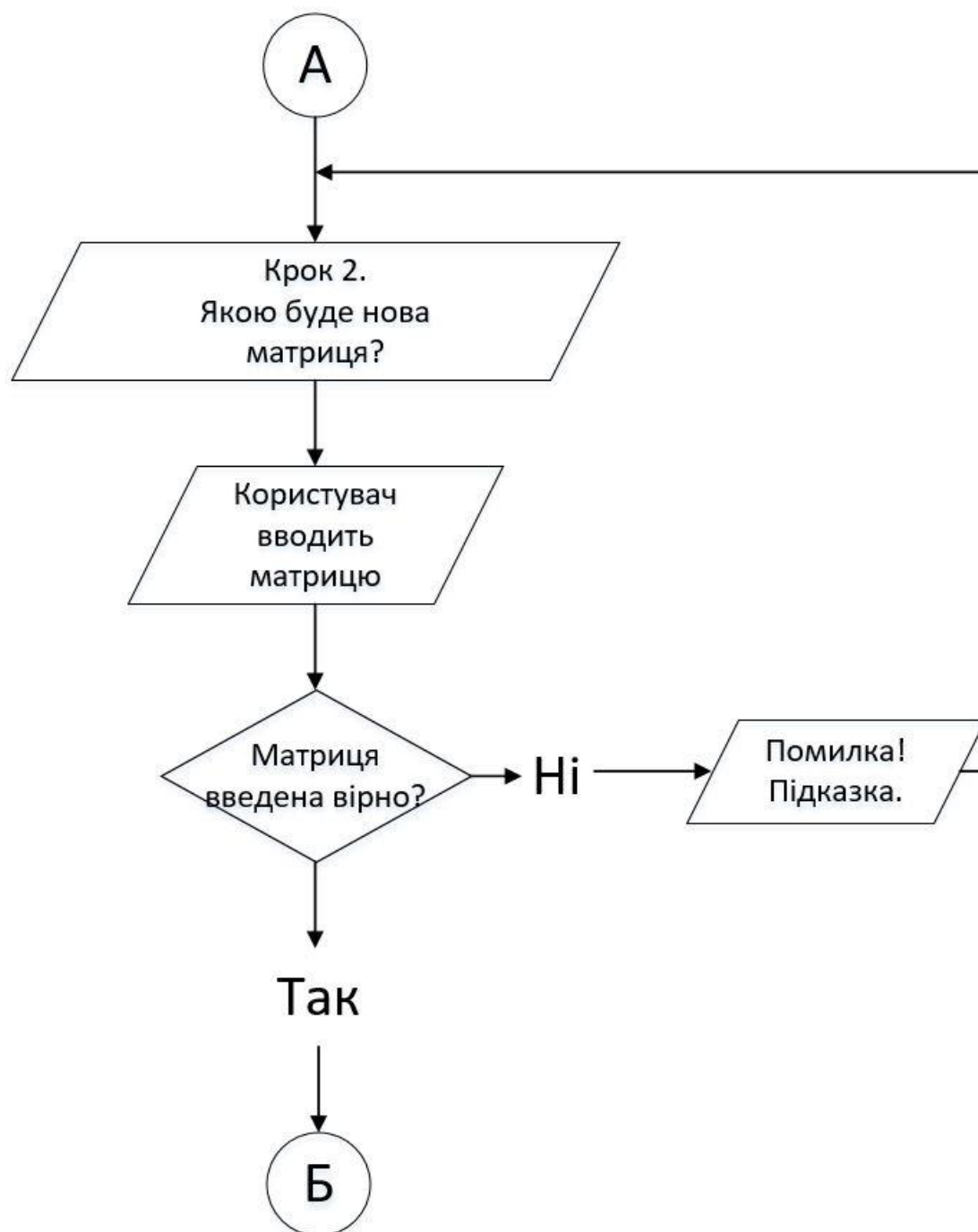


Рисунок 3.2 – Продовження блок-схеми алгоритму

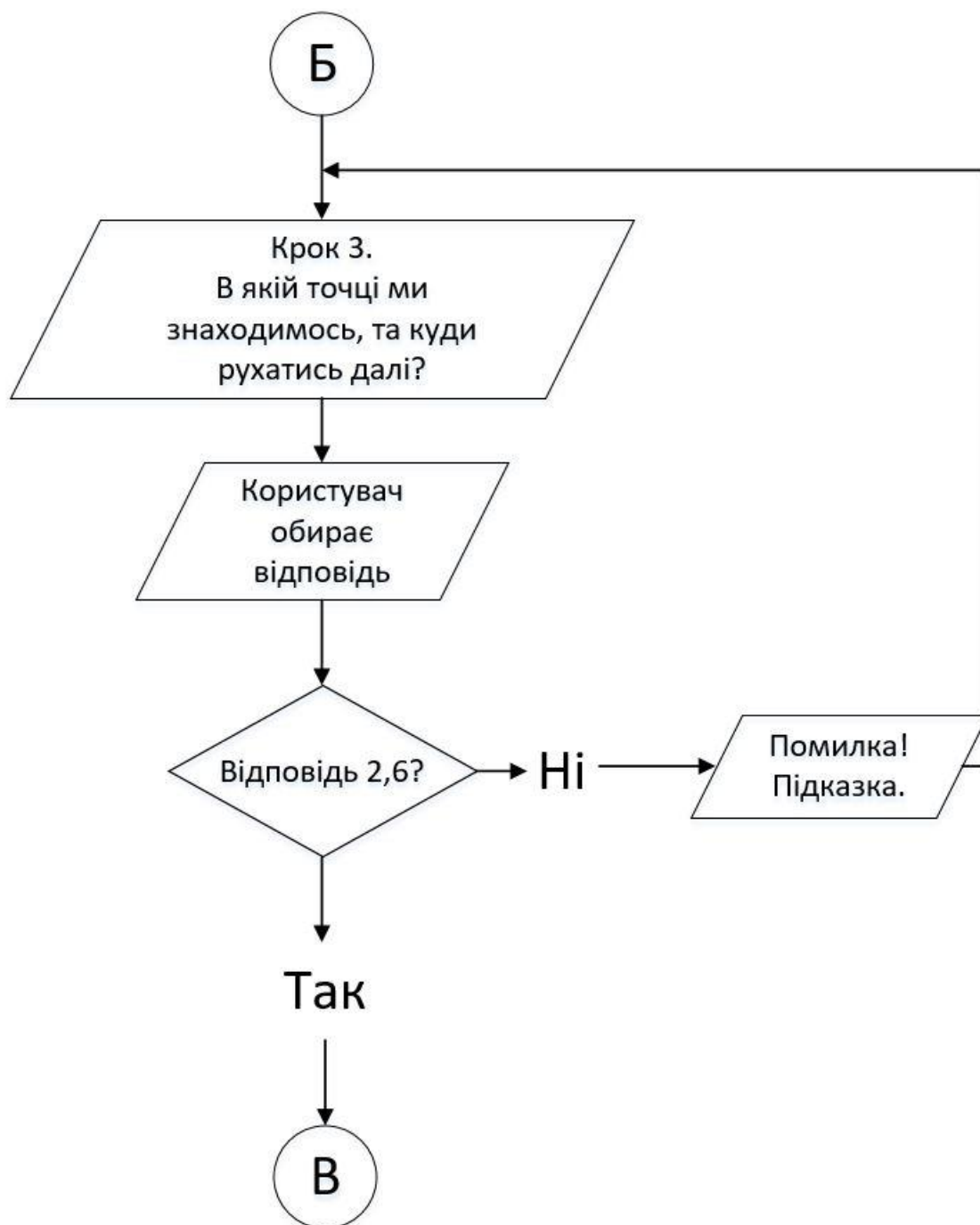


Рисунок 3.3 – Продовження блок-схеми алгоритму

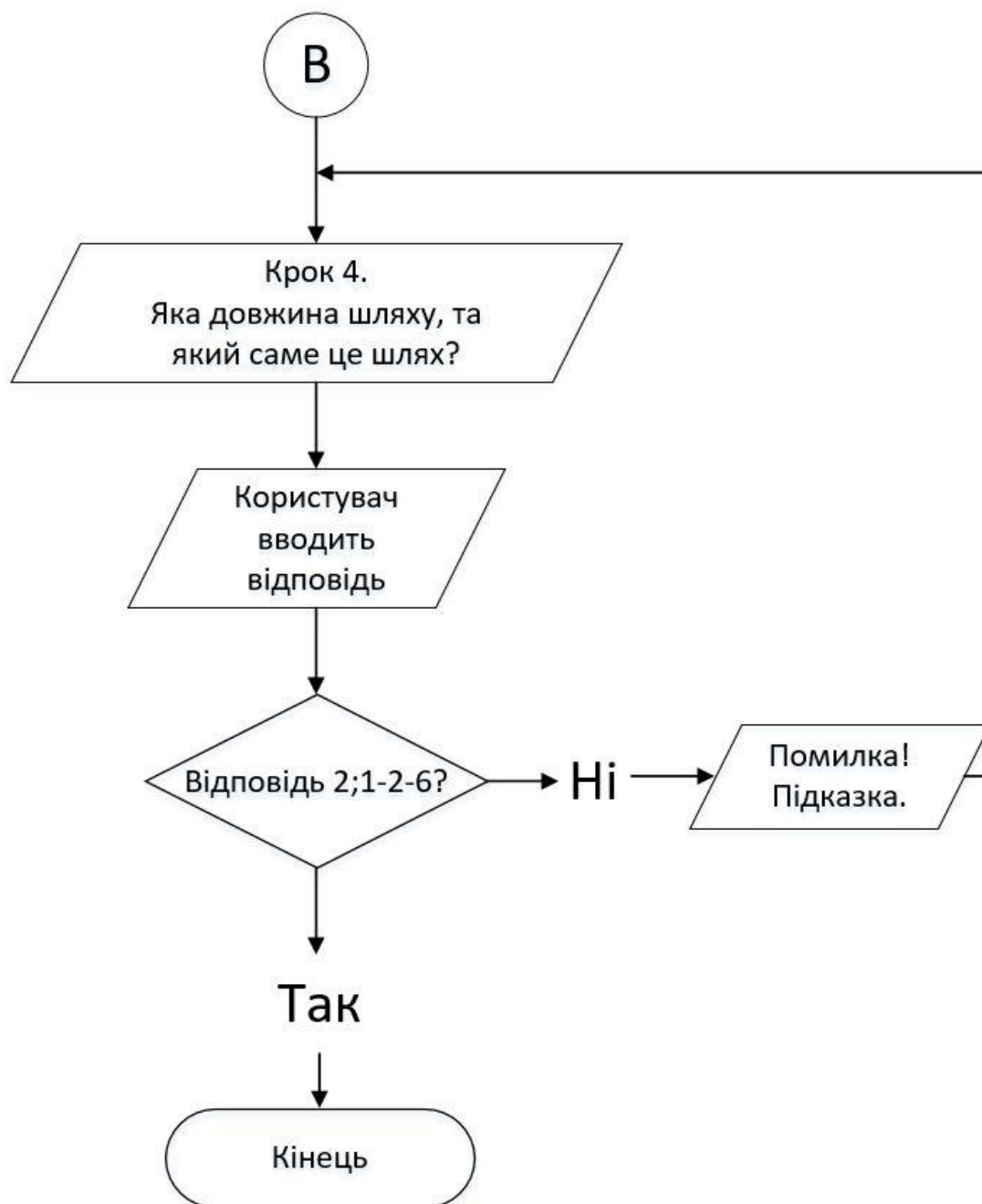


Рисунок 3.4 – Продовження блок-схеми алгоритму

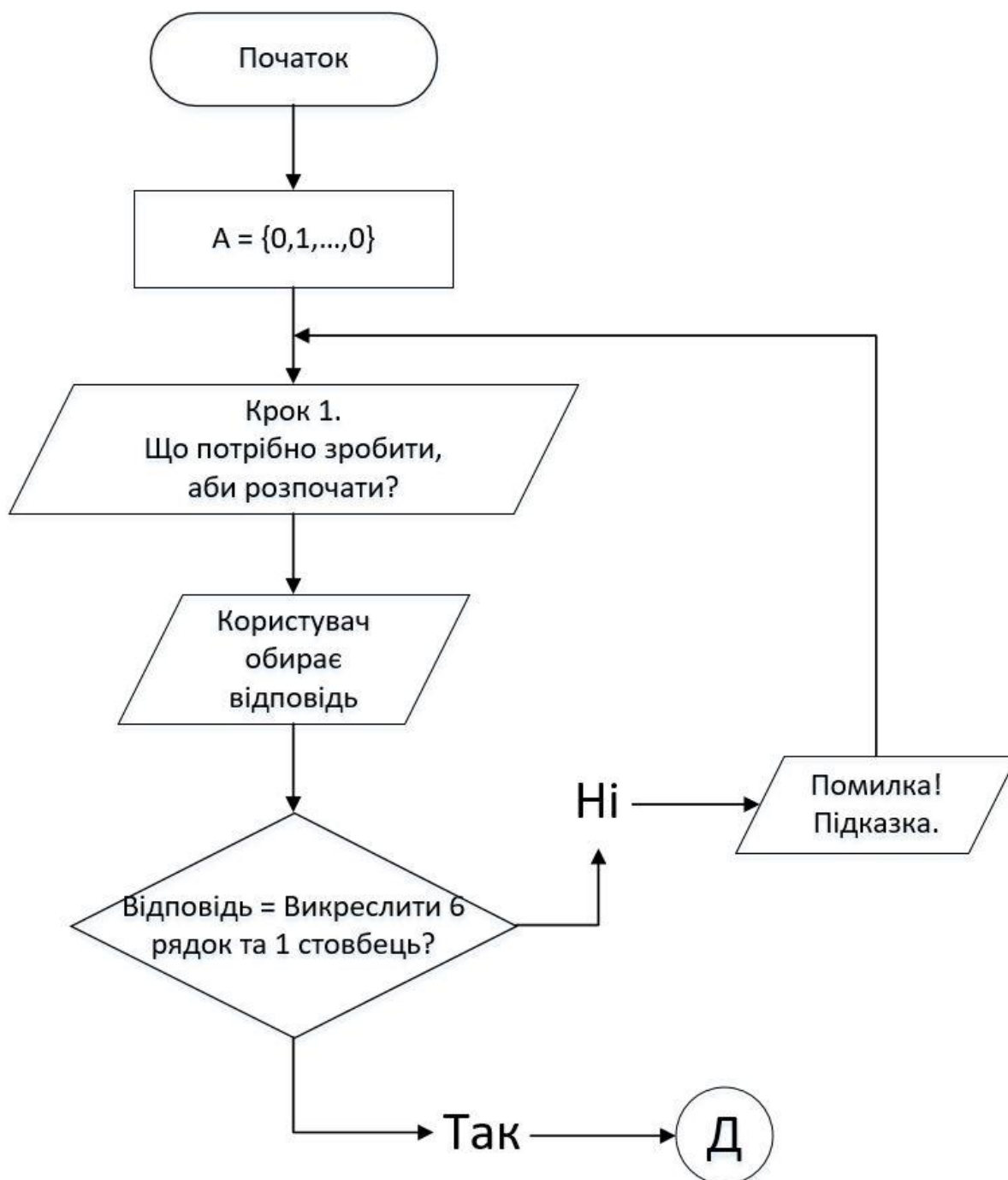


Рисунок 3.5 – Блок-схема алгоритму (тест2)

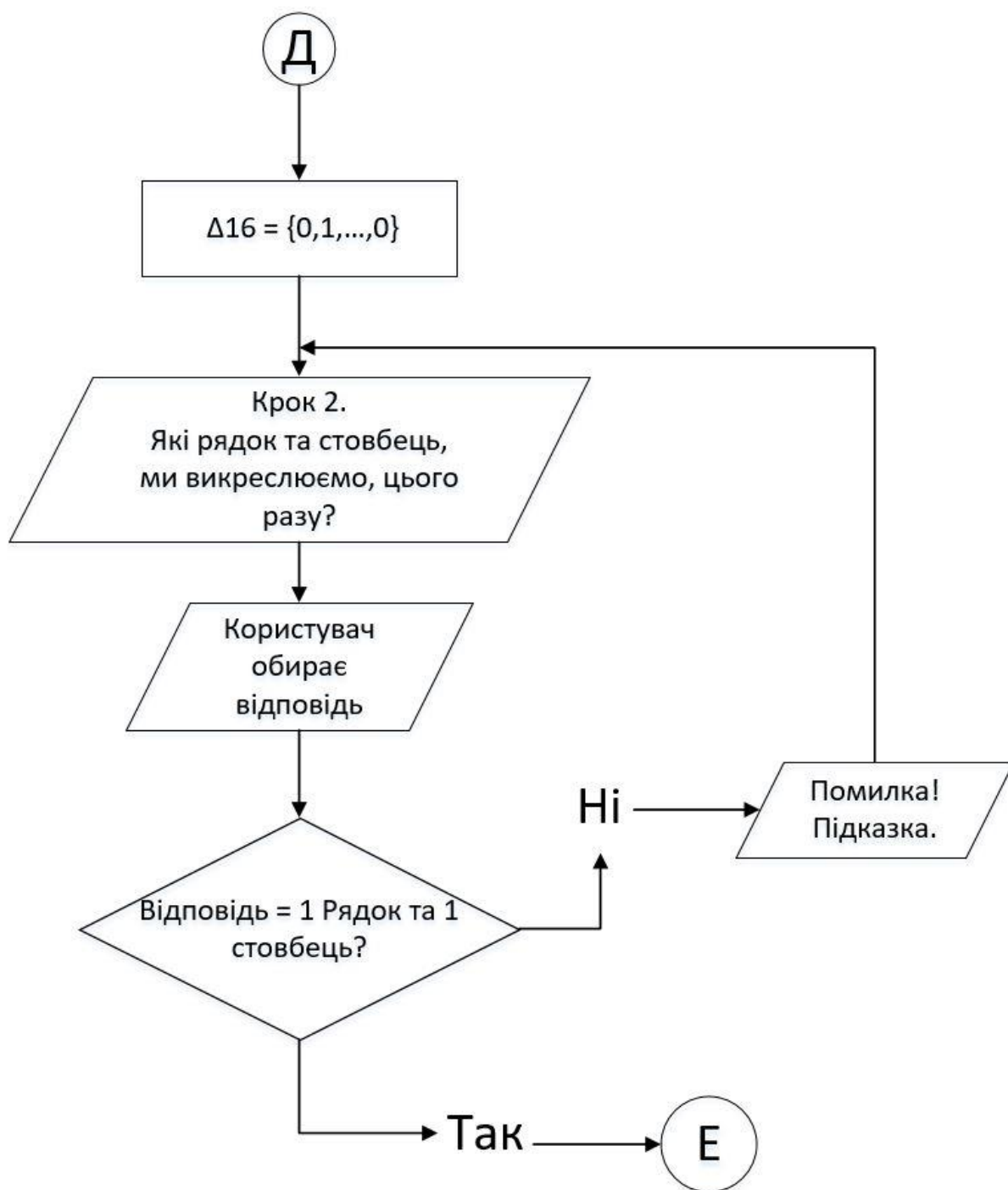


Рисунок 3.6 – Продовження блок-схеми алгоритму

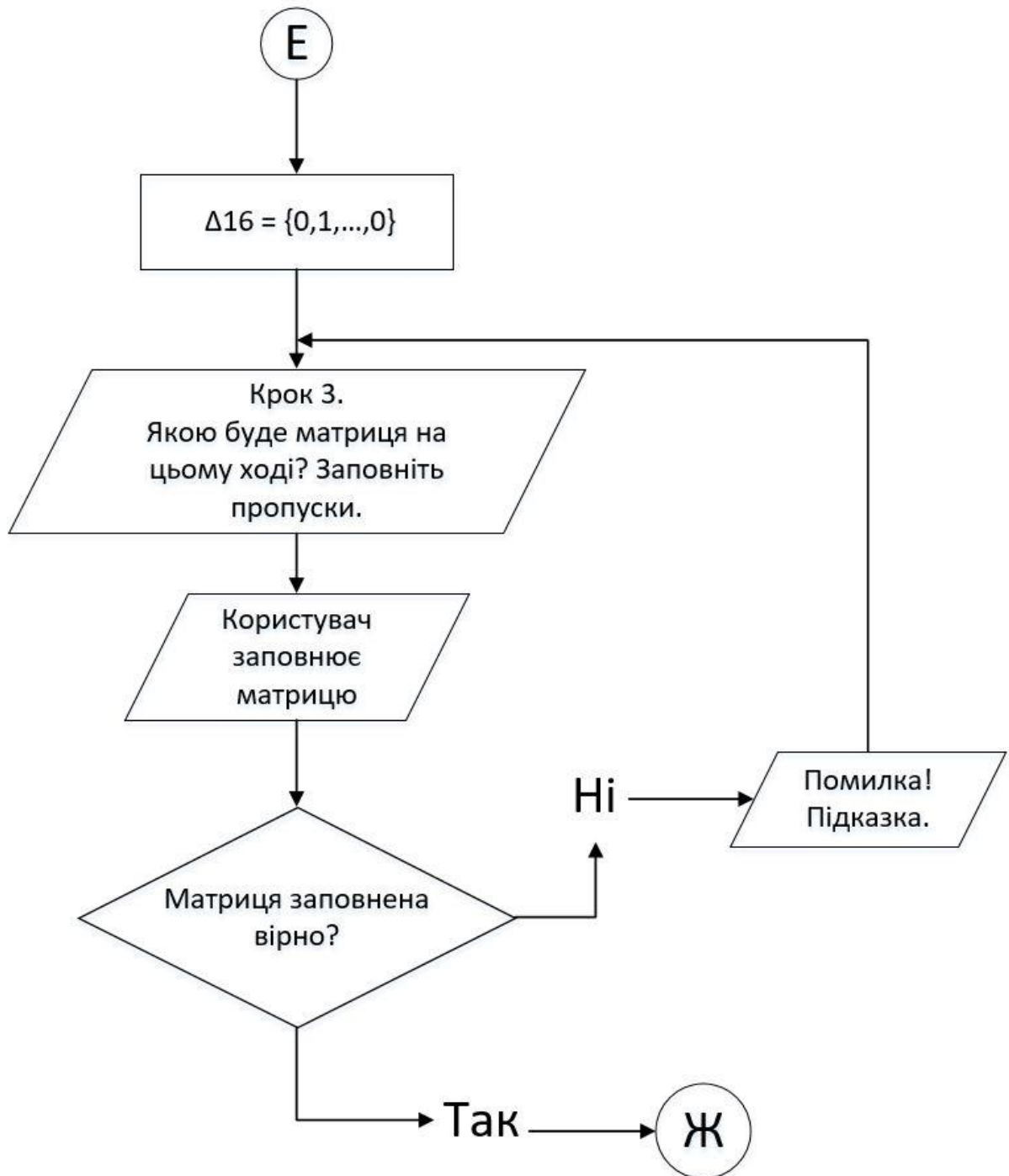


Рисунок 3.7 – Продовження блок-схеми алгоритму

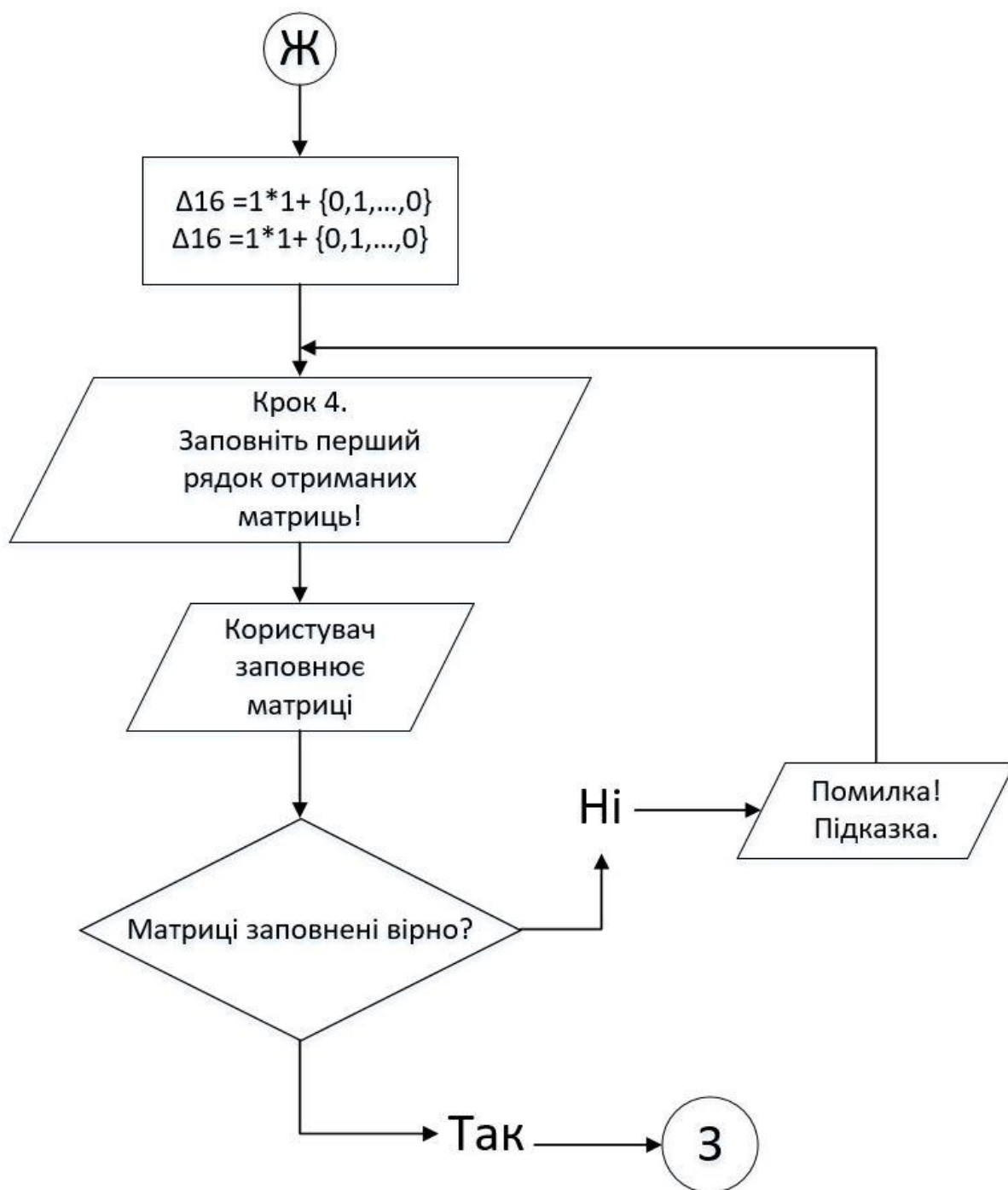


Рисунок 3.8 – Продовження блок-схеми алгоритму

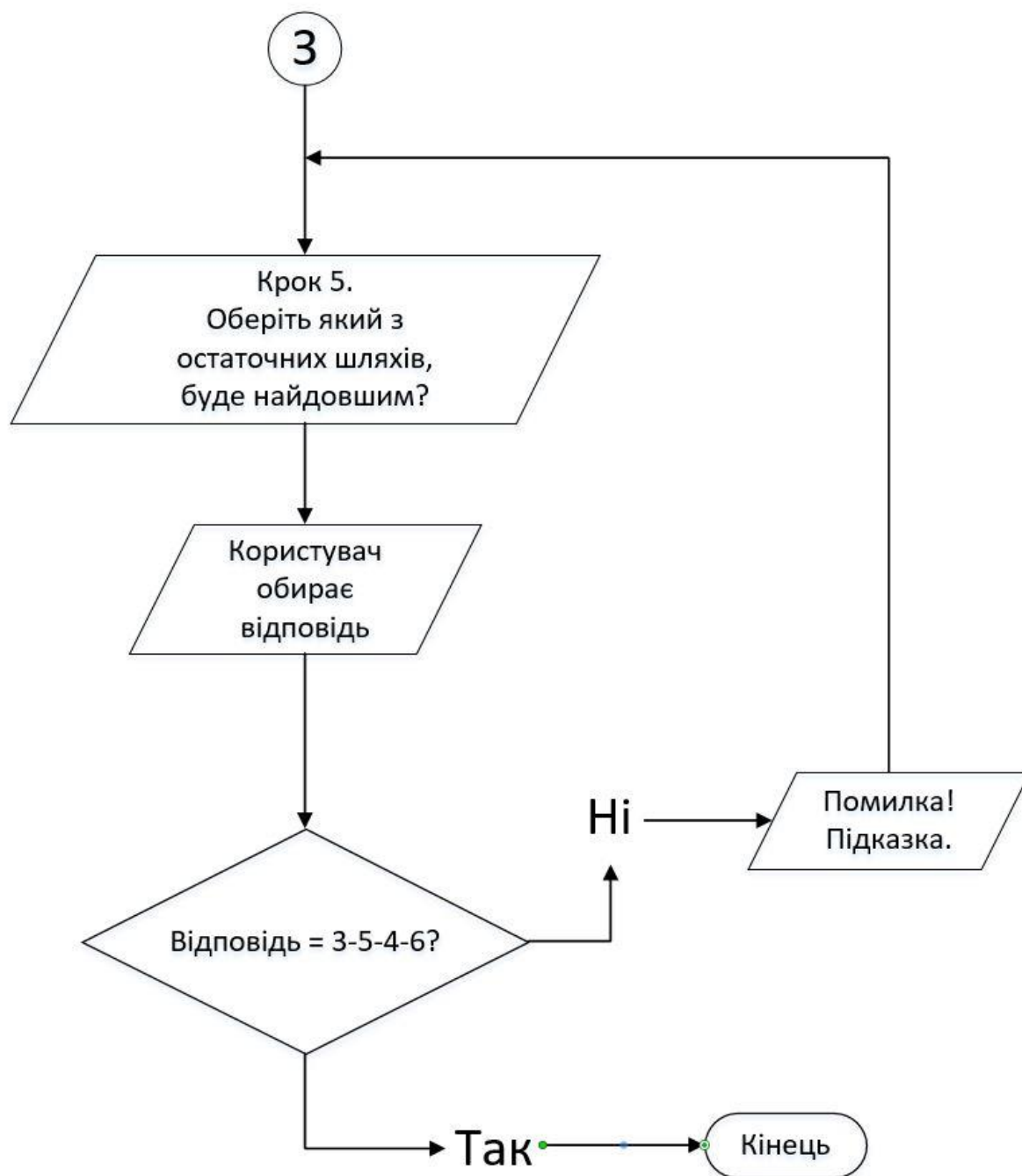


Рисунок 3.9 – Продовження блок-схеми алгоритму

4. ПРАКТИЧНА ЧАСТИНА

4.1 Інструкція по роботі з програмою

На рис. 4.1-4.31, продемонстровано основні етапи роботи програми.

1 Рівень:

Короткий шлях. Дано матрицю 6х6. Знайти найкоротший шлях з точки 1 в точку 6

Перший крок:

Знайти найкоротший шлях з вершини 1 у вершину 6

A=

index1	1*	2	3	4	5	6
▶ 1	0	1	0	0	0	0
2	0	0	1	1	0	1
3	0	0	0	1	1	0
4	0	0	0	0	0	1
5	0	0	0	1	0	1
6*	0	0	0	0	0	0

Який рядок та стовбець потрібно викреслити, аби розпочати?

☐ 2,5
 ☐ 5,6

☐ 6,1
 ☐ 1,2

Назад

Уперед

Рисунок 4.1 – Крок 1

Оберіть правильну відповідь.

Правильна відповідь : 6,1. Оскільки для початку нам потрібно викреслити стовбець з якого ми ідемо та рядок у який ми ідемо.

Другий крок:

Знайти найкоротший шлях з вершини 1 у вершину 6
×

A=

index1	1*	2	3	4	5	6
▶ 1	0	1	0	0	0	0
2	0	0	1	1	0	1
3	0	0	0	1	1	0
4	0	0	0	0	0	1
5	0	0	0	1	0	1
6*	0	0	0	0	0	0

δ16 =

	2	3	4	5	6*
▶ 1*					
2					
3					
4					
5					

Добре! Ідемо далі. Якою буде нова матриця?

Назад

Уперед

Рисунок 4.2 – Крок 2

Заповніть матрицю.

Правильна відповідь на рис. 4.3. Потрібно просто перенести значення з першої матриці у другу, без першого стовбця та останнього рядка.

Знайти найкоротший шлях з вершини 1 у вершину 6

×

index1

1*

2

3

4

5

6

▶ 1

0

1

0

0

0

0

2

0

0

1

1

0

1

3

0

0

0

1

1

0

4

0

0

0

0

0

1

5

0

0

0

1

0

1

6*

0

0

0

0

0

0

A=

index1	1*	2	3	4	5	6
▶ 1	0	1	0	0	0	0
2	0	0	1	1	0	1
3	0	0	0	1	1	0
4	0	0	0	0	0	1
5	0	0	0	1	0	1
6*	0	0	0	0	0	0

2

3

4

5

6*

1*

1

0

0

0

0

2

0

1

1

0

1

3

0

0

1

1

0

4

0

0

0

0

1

5

0

0

1

0

1

δ16 =

	2	3	4	5	6*
1*	1	0	0	0	0
2	0	1	1	0	1
3	0	0	1	1	0
4	0	0	0	0	1
5	0	0	1	0	1

Добре! Ідемо далі. Якою буде нова матриця?

Назад

Уперед


Рисунок 4.3 – Правильна відповідь до кроку 2

Як бачимо, маємо в поточній матриці лише один єдиний крок з вершини 1 у вершину 2.

Переходячи по ньому, довжина шляху збільшується на 1, та з'являється нова матриця.

Виходячи з цього, переходимо до 3 кроку.

Третій крок:

 Знайти найкоротший шлях з вершини 1 у вершину 6
 ✕

$\delta_{16} =$

	2	3	4	5	6*
1*	1	0	0	0	0
2	0	1	1	0	1
3	0	0	1	1	0
4	0	0	0	0	1
▶ 5	0	0	1	0	1

$= 1 +$

	3	4	5	6*
▶ 2*	1	1	0	1
3	0	1	1	0
4	0	0	0	1
5	0	1	0	1

В якій точці ми знаходимося, та в яку точку нам потрібно рухатися далі?

☐ 2,5
 ☐ 4,5

☐ 3,6
 ☒ 2,6

Назад

Уперед

Рисунок 4.4 – Крок 3

Оберіть правильну відповідь.

Правильна відповідь : 2,6. На попередньому кроці, ми перейшли з вершини 1 у вершину 2(оскільки це був єдиний можливий варіант). А рухатися нам потрібно у вершину 6, оскільки це мета поточного тесту(знаходження найкоротшого шляху). Якщо ми підемо у інші вершини, від цього не буде ніякого сенсу, адже їх довжина на поточному кроці, вже буде дорівнювати довжині найкоротшого шляху.

Четвертий крок:

Знайти найкоротший шлях з вершини 1 у вершину 6

×

2

3

4

5

6*

1*

2

3

4

▶ 5

1

0

0

0

0

0

1

1

0

1

0

0

1

1

0

0

0

0

0

1

0

0

1

0

1

3

4

5

6*

▶ 2*

3

4

5

1

1

0

1

0

1

1

0

0

0

0

1

0

1

0

1

Чудово! Ми перейшли із вершини 2 у вершину 6. Познач яка довжина шляху який ми пройшли, та який саме це шлях?

Приклад (4; 3-5-1)

Довжина шляху:

2

Шлях:

1-2-6

Назад

Уперед

Рисунок 4.5 – Крок 4

Оберіть правильну відповідь.

Правильна відповідь : Викреслити 6 рядок та 1 стовбець. Аналогічно до попереднього тесту для початку нам потрібно викреслити стовбець з якого ми ідемо та рядок у який ми ідемо.

Другий крок:

Знайти найдовший шлях з вершини 1 у вершину 6

×

index2

2

3

4

5

6*

▶ 1*

1

0

0

0

0

2

0

1

1

0

1

3

0

0

1

1

0

4

0

0

0

0

1

5

0

0

1

0

1

Δ16 =

▶ 1*

1

0

0

0

0

Які рядок та стовбець, ми викреслюємо, цього разу?

☐ 2 Рядок та 1 стовбець

☐ 1 Рядок та 2 стовбець

☒ 1 Рядок та 1 стовбець

☐ 2 Рядок та 2 стовбець

Назад

Уперед

Рисунок 4.7 – Крок 2

Оберіть правильну відповідь.

Заповнити матрицю.

index3	3	4	5	6*
2*	1	1	0	1
3	0	1	1	0
4	0	0	0	1
5	0	1	0	1

Рисунок 4.9 – Заповнена матриця кроку 3

Правильна відповідь : Оскільки на попередньому кроці, ми перейшли з вершини 1, у вершину 2, і викреслили 1 рядок та 1 стовбець, отримали нову матрицю. Аби її заповнити, потрібно просто перенести усі елементи, які в нас залишилися з попередньої матриці, коли ми викреслили перший рядок та перший стовпець (рис. 4.9).

Четвертий крок:

Знайти найдовший шлях з вершини 1 у вершину 6

$\Delta 16 = 1 *$

index3	3	4	5	6*
2*	1	1	0	1
3	0	1	1	0
4	0	0	0	1
5	0	1	0	1

$\Delta 16 = 1 * 1 +$

index4	4	5	6*
3*		1	0
4		0	1
5		0	1

$\Delta 16 = 1 * 1 +$

index4	3	5	6*
3		1	0
4*		0	1
5		0	1

Заповніть перший рядок отриманих матриць! Попередня матриця знаходиться з лівої частини.Зверніть увагу - Зтя можлива матриця зі шляхом 1-2-6, має найкоротший шлях (висновок з попереднього тесту), тому її ми враховувати не будемо.

Назад

Уперед

Рисунок 4.10 – Крок 4

Заповнити матриці.

Правильна відповідь: Знаходячись у вершині 2, маємо 3 можливі шляхи: перший піти у вершину 3, другий у вершину 4, третій у вершину 6. Оскільки, іти у вершину 6 немає ніякого сенсу(найкоротший шлях з попереднього тестування), спробуємо зрозуміти який із 2х шляхів що нам залишилися – найдовший. Одразу сказати який з них буде довшим дуже важко, тому спробуємо їх порівняти, для цього підемо одразу в обидві вершини. На цьому етапі, користувач вчиться заповнювати одразу 2 матриці, оскільки це різні шляхи, то і матриці мають бути заповнені по різному (рис. 4.11).

Знайти найдовший шлях з вершини 1 у вершину 6

$\Delta 16 = 1 *$

index3	3	4	5	6*
2*	1	1	0	1
3	0	1	1	0
4	0	0	0	1
▶ 5	0	1	0	1

$\Delta 16 = 1 * 1 +$

index4	4	5	6*
3*	1	1	0
4	0	0	1
▶ 5	1	0	1

$\Delta 16 = 1 * 1 +$

index4	3	5	6*
3	0	1	0
4*	0	0	1
▶ 5	0	0	1

Заповніть перший рядок отриманих матриць! Попередня матриця знаходиться з лівої частини. Зверніть увагу - Зтя можлива матриця зі шляхом 1-2-6, має найкоротший шлях (висновок з попереднього тесту), тому її ми враховувати не будемо.

Назад
Уперед

Рисунок 4.11 – Заповнені матриці кроку 4

Висновок: Після виконання цього завдання, користувачу до уваги буде надано діалогове вікно (рис. 4.12), в якому є відповідь, чому нам не підходить 2 шлях. Тому ми беремо до уваги лише 1 перший шлях. На цьому шляхові ми знаходимось у вершині 3(Можна зрозуміти по * коло індексу рядка).

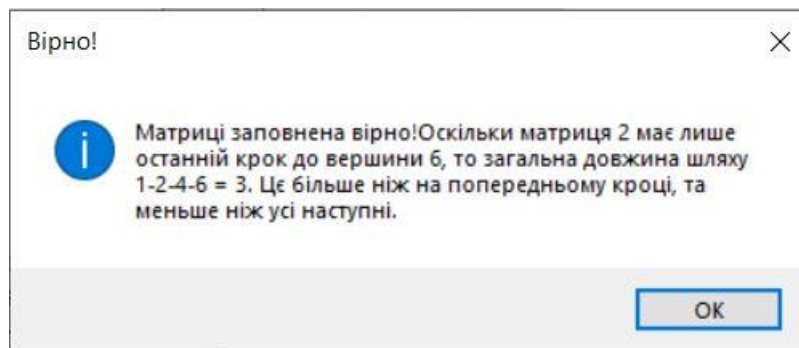


Рисунок 4.12 – Підказка після кроку 4

П'ятий крок:

Знайти найдовший шлях.

Правильна відповідь: 3-5-4-6 (рис. 4.13). Знаходячись у вершині 3, бачимо що знову є 2 доступних шляхи. На цьому етапі, користувач може усно відшукати усі ймовірні шляхи, та порівняти їх із собою. Шукаючи шляхи усно, можна швидко зрозуміти, що надані нижче варіанти(рис. 4.13), це усі можливі шляхи. Шлях до цього етапу = 2. Порівняючи між собою усі надані варіанти, можна швидко дійти висновку, що найдовшим буде той, у якого найдовший шлях. В даному випадку шляхи 3-5-6 та 3-4-6 будуть = 4, а шлях 3-5-4-6 = 5, тому обираємо його. Після завершення тесту, користувач побачить діалогове вікно, з усіма можливими шляхами.

Знайти найдовший шлях з вершини 1 у вершину 6

$\Delta 16 = 1 * 1 +$

index4	4	5	6*
3*	1	1	0
4	0	0	1
5	1	0	1

Добре!! Оскільки матриця 2 має лише останній крок до вершини 6, то загальна довжина шляху 1-2-4-6 = 3. Це більше ніж на попередньому кроці, та менше ніж усі наступні. Оберіть який з остаточних шляхів на вашу думку, буде найдовшим?

☐ 3-5-6
☐ 3-5-4-6
☐ 3-4-6

Назад

Уперед

Рисунок 4.13 – Крок 5

Заповніть матрицю по малюнку.

Правильна відповідь: На рис. 4.14 зображено граф. Користувач повинен заповнити матрицю по малюнку, вписуючи 1, якщо шлях з однієї вершини у іншу можливий, та 0 – якщо шляху не має (рис. 4.15).

Короткий шлях (2 Рівень)

A =

Index5	1	2	3	4	5*
1*	0	1	1	1	0
2	0	0	0	1	1
3	0	0	0	0	1
4	0	0	1	0	1
5	1	0	0	0	0

Заповніть матрицю по малюнку, використовуючи 1 - для відображення доступного шляху, та 0 - для відображення відсутності шляху

Вперед

Рисунок 4.15 – Заповнена матриця 1 кроку

Другой krok:

Короткий шлях (2 Рівень)
✕

A =

Index5	1	2	3	4	5*
▶ 1*	0	4	2	3	0
2	0	0	0	1	2
3	0	0	0	0	3
4	0	0	2	0	1
5	2	0	0	0	0

δ15 = 2 +

Index4	1	2	4	5*
▶ 2				
3*				
4				
5				

Знайдіть найкоротший шлях, з вершини 1 у вершину 5.
 Пропоную розпочати з вершини 3, оскільки в цьому стовбчику - найменший елемент в рядочку. Впишіть, яка матриця вийде, якщо перейти з вершини 1 у вершину 3?
 Зверніть увагу, що ваги шляхів вже не 1 та 0!!

Вперед

Рисунок 4.16 – Крок 2

Заповніть матрицю.

Правильна відповідь: Оскільки ми не можемо одразу вказати найкоротший шлях, через потрібне розгалуження, то спробуємо спочатку піти у ту вершину, елемент якої є найменшим у рядкові. Такий елемент знаходиться у вершині 3. Тож користувачу потрібно ввести, яка матриця вийде, якщо перейти у вершину 3. Враховуючи, що тепер індекси не тільки 0 та 1 (рис. 4.17).

Короткий шлях (2 Рівень)

Index5

	1	2	3	4	5*
1*	0	4	2	3	0
2	0	0	0	1	2
3	0	0	0	0	3
4	0	0	2	0	1
5	2	0	0	0	0

Index4

	1	2	4	5*
2	0	0	1	2
3*	0	0	0	3
4	0	0	0	1
5	2	0	0	0

$\delta_{15} = 2 +$

Знайдіть найкоротший шлях, з вершини 1 у вершину 5.
 Пропоную розпочати з вершини 3, оскільки в цьому стовбчику - найменший елемент в рядочку. Впишіть, яка матриця вийде, якщо перейти з вершини 1 у вершину 3?
 Зверніть увагу, що ваги шляхів вже не 1 та 0!!

Вперед

Рисунок 4.17 – Заповнена матриця 2 кроку

Третій крок:

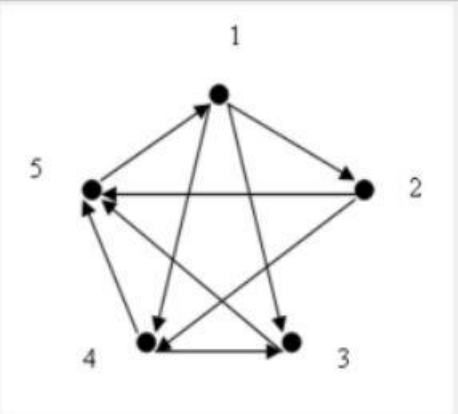
Короткий шлях (2 Рівень)

A =

Index5	1	2	3	4	5*
► 1*	0	4	2	3	0
2	0	0	0	1	2
3	0	0	0	0	3
4	0	0	2	0	1
5	2	0	0	0	0

δ15 = 2 +

Index4	1	2	4	5*
2	0	0	1	2
3*	0	0	0	3
4	0	0	0	1
► 5	2	0	0	0



Якою буде сума цього шляху?

☐ 4 ☐ 5
☐ 3 ☐ 7

Вперед

Рисунок 4.18 – Крок 3

Обрати відповідь.

Правильна відповідь: 5. З матриці яка в нас вийшла, одразу можна побачити, що єдина вершина, у яку ми можемо потрапити з вершини 3 – це вершина 5. А порахувавши суму елементів, які зустрічаються нам протягом шляху (з вершини 1 у вершину 3 = 2, а з вершини 3 у вершину 5 = 3; $2+3 = 5$), можна не вагаючись обрати відповідь.

Четвертий крок:

Короткий шлях (2 Рівень)

×

Index5

1

2

3

4

5*

▶ 1*

2

3

4

5

0

4

2

3

0

0

0

0

0

2

0

0

0

0

3

0

0

2

0

1

2

0

0

0

0

1

2

3

4

5

Index4

1

2

3

5*

▶ 2

3

4*

5

815 = 3 +

Яка матриця вийде, якщо перейти з вершини 1 у вершину 4?

Вперед

Рисунок 4.19 – Крок 4

Заповнити матрицю.

Правильна відповідь: Оскільки те що ми знайшли яка довжина шляху 1-3-5, не дає нам гарантованої відповіді, підемо у наступну вершину, елемент якої є наступним найменшим елементом рядка. Це вершина 4. Тож потрібно заповнити матрицю, яка вийде, якщо ми підемо у вершину 4 (рис. 4.20).

Короткий шлях (2 Рівень)

×

Index5

1

2

3

4

5*

► 1*

2

3

4

5

0

4

2

3

0

0

0

0

1

2

0

0

0

0

3

0

0

2

0

1

2

0

0

0

0

1

2

3

4

5

Index4

1

2

3

5*

2

3

4*

5

0

0

0

2

0

0

0

3

0

0

2

1

2

0

0

0

815 = 3 +

Яка матриця вийде, якщо перейти з вершини 1 у вершину 4?

Вперед

Рисунок 4.20 – Заповнена матриця кроку 4

П'ятий крок:

Короткий шлях (2 Рівень)

✕

Index5	1	2	3	4	5*
► 1*	0	4	2	3	0
2	0	0	0	1	2
3	0	0	0	0	3
4	0	0	2	0	1
5	2	0	0	0	0

A =

Index4	1	2	3	5*
2	0	0	0	2
3	0	0	0	3
4*	0	0	2	1
► 5	2	0	0	0

$\delta_{15} = 3 +$

Чому буде дорівнювати пройдений шлях, коли ми потрапимо у вершину 5 з вершини 4?

Відповідь запишіть цифрою.

Подумайте, чи варто нам іти з вершини 1 у вершину 2.

Шлях :

Вперед

Рисунок 4.21 – Крок 5

Вписати довжину шляху.

Правильна відповідь: 4. Перехід з вершини 1 в 4 = 3. Перехід з вершини 4 у вершину 5 = 1. $3+1 = 4$. Отже довжина шляху 1-4-5 = 4.

Шостий крок:

Короткий шлях (2 Рівень)

Index5	1	2	3	4	5*
1*	0	4	2	3	0
2	0	0	0	1	2
3	0	0	0	0	3
4	0	0	2	0	1
5	2	0	0	0	0

A =

Index4	1	2	3	5*
2	0	0	0	2
3	0	0	0	3
4*	0	0	2	1
5	2	0	0	0

δ15 = 3 +

Оберіть найкоротший із запропонованих шляхів.

☐ 1-3-5 ☐ 1-2-4-5
☐ 1-2-5 ☐ 1-4-5

Вперед

Рисунок 4.22 – Крок 6

Оберіть правильну відповідь.

Правильна відповідь: 1-4-5. Ми це дізналися з попереднього кроку. Оскільки перехід у вершину 3, дає нам довжину 5, а довжина обраного шляху =

4, то вершина 3 одразу відпадає. Перехід у вершину 2, взагалі не має сенсу розглядати, оскільки лише перехід туди буде дорівнювати довжині шляху 1-4-5.

Шлях 1-4-3-5, також відпадає, оскільки це той же шлях що ми обрали, але с додатковим переходом.

2 Рівень.

Найдовший шлях. Дано граф з п'ятьма вершинами. Знайти найдовший шлях з вершини 1 у вершину 5

Перший крок:

$A =$

	1*	2	3	4	5
1	0	-4	-2	-3	0
2	0	0	0	-1	-2
3	0	0	0	0	-3
4	0	0	-2	0	-1
5*	-2	0	0	0	0

$\Delta 15 =$

	2	3	4	5*
1*				
2				
3				
4				

Заповніть матрицю $\|D15\|$

Відповісти

Рисунок 4.23 – Крок 1

Заповніть матрицю.

Правильна відповідь: Для того, аби почати, потрібно заповнити матрицю $\|D15\|$ для знаходження найдовшого шляху. Для цього спочатку викреслюємо

перший стовбець, та останній рядок. Далі заповнюємо матрицю тими елементами, які залишаться (рис. 4.24).

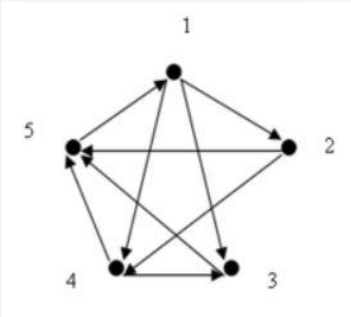
$\Delta 15 =$

	2	3	4	5*
1*	-4	-2	-3	0
2	0	0	-1	-2
3	0	0	0	-3
4	0	-2	0	-1

Рисунок 4.24 – Заповнена матриця з кроку 1

Другий крок:

Довгий шлях (2 Рівень)



$A =$

	1*	2	3	4	5
1	0	-4	-2	-3	0
2	0	0	0	-1	-2
3	0	0	0	0	-3
4	0	0	-2	0	-1
5*	-2	0	0	0	0

$\Delta 15 =$

	2	3	4	5*
1*	-4	-2	-3	0
2	0	0	-1	-2
3	0	0	0	-3
4	0	-2	0	-1

$\Delta 15 = (-4) + A12$

	3	4	5*
2*			
3			
4			

$\Delta 15 = (-3) + A14$

	2	3	5*
2			
3			
4*			

Яка довжина шляху 1-3-5?

Довжина :

Відповісти

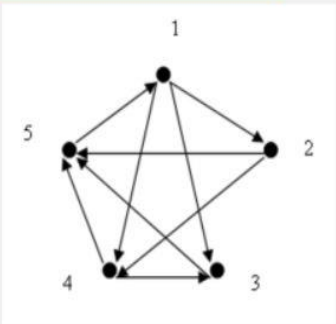
Рисунок 4.25 – Крок 2

Заповнити матриці, та вписати довжину.

Правильна відповідь: Довжина буде = 5. Оскільки перехід з вершини 1 у вершину 3 = -2, а перехід з вершини 3 у 5 = -3. $|-2 -3| = 5$. Отже довжина першого шляху нам відома. Спробуємо знайти довжини інших 2х шляхів.

Перший шлях буде через вершину 2, а другий через вершину 4. Отже в відповідних матрицях викреслюємо перший рядок та перший/третій стовбці відповідно(2 та 4 вершинам). Записуємо матриці, які в нас залишилися(рис. 4.26).

Довгий шлях (2 Рівень)



Matrix A:

	1*	2	3	4	5
1	0	-4	-2	-3	0
2	0	0	0	-1	-2
3	0	0	0	0	-3
4	0	0	-2	0	-1
5*	-2	0	0	0	0

$\Delta 15 =$

	2	3	4	5*
1*	-4	-2	-3	0
2	0	0	-1	-2
3	0	0	0	-3
4	0	-2	0	-1

$\Delta 15 = (-4) + A12$

	3	4	5*
2*	0	-1	-2
3	0	0	-3
4	-2	0	-1

$\Delta 15 = (-3) + A14$

	2	3	5*
2	0	0	-2
3	0	0	-3
4*	0	-2	-1

Яка довжина шляху 1-3-5?

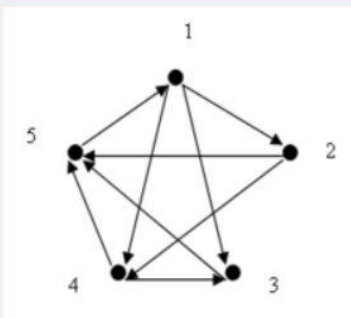
Довжина :

Відповісти

Рисунок 4.26 – Заповнені матриці з кроку 2

Третій крок:

Довгий шлях (2 Рівень)



$A =$

	1*	2	3	4	5
1	0	-4	-2	-3	0
2	0	0	0	-1	-2
3	0	0	0	0	-3
4	0	0	-2	0	-1
5*	-2	0	0	0	0

$\Delta 15 =$

	2	3	4	5*
1*	-4	-2	-3	0
2	0	0	-1	-2
3	0	0	0	-3
4	0	-2	0	-1

$\Delta 15 = (-4) + A12$

	3	4	5*
2*	0	-1	-2
3	0	0	-3
4	-2	0	-1

 $= (-4) * \min \{ (-1) + A24$

	3	5*
3		
4*		

 $+ (-2) \}$

$\Delta 15 = (-3) + A14$

	2	3	5*
2	0	0	-2
3	0	0	-3
4*	0	-2	-1

 $= (-3) * \min \{ (-2) + A43$

	2	5*
2		
3*		

 $+ (-1) \}$

Маємо 2 шляхи, продовжіть заповнювати матриці.
Зверніть увагу, у матрицях зірочка стоїть біля тієї вершини, у якій ми знаходимось!

Відповісти

Рисунок 4.27 – Крок 3

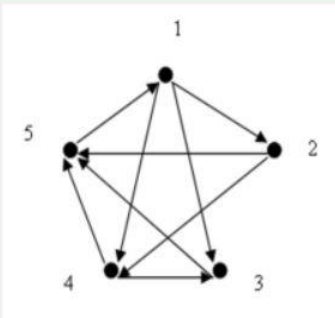
Заповнити матриці.

Правильна відповідь:

У першому шляху потрібно заповнити матрицю з переходом з другої вершини у 4ту, оскільки перехід одразу у вершину 5, буде мати меншу довжину шляху, а отже не дасть нам бажаного результату(найдовший шлях). Тож потрібно заповнити матрицю, яка вийде після переходу у вершину 4.

У другому шляху ми пішли у вершину 3, оскільки перехід одразу у вершину 5, буде мати меншу довжину шляху, а отже не дасть нам бажаного результату(найдовший шлях). Тож потрібно заповнити матрицю, яка вийде після переходу у вершину 3 (рис. 4.28).

Довгий шлях (2 Рівень)



$A =$

	1*	2	3	4	5
▶ 1	0	-4	-2	-3	0
2	0	0	0	-1	-2
3	0	0	0	0	-3
4	0	0	-2	0	-1
5*	-2	0	0	0	0

$\Delta 15 =$

	2	3	4	5*
1*	-4	-2	-3	0
2	0	0	-1	-2
3	0	0	0	-3
▶ 4	0	-2	0	-1

$\Delta 15 = (-4) + A12$

	3	4	5*
2*	0	-1	-2
3	0	0	-3
▶ 4	-2	0	-1

$= (-4) * \min \{ (-1) + A24$

	3	5*
3	0	-3
▶ 4*	-2	-1

$+ (-2) \}$

$\Delta 15 = (-3) + A14$

	2	3	5*
2	0	0	-2
3	0	0	-3
▶ 4*	0	-2	-1

$= (-3) * \min \{ (-2) + A43$

	2	5*
2	0	-2
▶ 3*	0	-3

$+ (-1) \}$

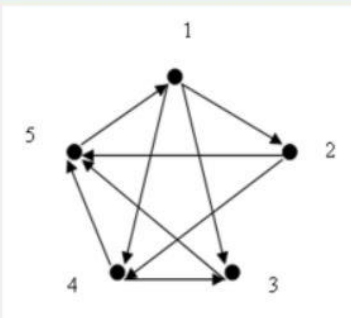
Маємо 2 шляхи, продовжіть заповнювати матриці.
Зверніть увагу, у матрицях зірочка стоїть біля тієї вершини, у якій ми знаходимось!

Відповісти

Рисунок 4.28 – Заповнені матриці з кроку 3

Четвертий крок:

Довгий шлях (2 Рівень)



$A =$

	1*	2	3	4	5
▶ 1	0	-4	-2	-3	0
2	0	0	0	-1	-2
3	0	0	0	0	-3
4	0	0	-2	0	-1
5*	-2	0	0	0	0

$\Delta 15 =$

	2	3	4	5*
1*	-4	-2	-3	0
2	0	0	-1	-2
3	0	0	0	-3
▶ 4	0	-2	0	-1

$\Delta 15 = (-4) + A12$

	3	4	5*
2*	0	-1	-2
3	0	0	-3
▶ 4	-2	0	-1

 $= (-4) * \min \{ (-1) + A24$

	3	5*
3	0	-3
▶ 4*	-2	-1

 $+ (-2) \} * (-2) * (-3)$

$\Delta 15 = (-3) + A14$

	2	3	5*
2	0	0	-2
3	0	0	-3
▶ 4*	0	-2	-1

 $= (-3) * \min \{ (-2) + A43$

	2	5*
2	0	-2
▶ 3*	0	-3

 $+ (-1) \} * (-3)$

$\Delta 15 = | \min \{ -4 + A12; -2 + A13; -3 + A14 \} | = | \min \{ -4 - 1 - 2 - 3; -2 - 3; -3 - 2 - 3 \} | = ??$ Чому дорівнює найдовший шлях?

Шлях :

Відповісти

Рисунок 4.29 – Крок 4

Знайти найдовший шлях.

Правильна відповідь: Користуючись формулами із лекції, можна з легкістю порахувати довжини шляхів, які ми пройшли.

Розпочнемо з нижнього шляху. Хронологія: Спочатку ми перейшли у вершину 4, довжина шляху = -3. Потім з вершини 4, ми вирішили піти у вершину 3, оскільки довжина шляху при переході у вершину 5 буде менша, аніж при переході у вершину 3. А з вершини 3, єдина вершина у яку ми можемо переміститися це вершина 5 (що нам і було потрібно по умові). Отже, довжина цього шляху $\Delta 15 = | -3 - 2 - 3 | = | -8 | = 8$.

Тепер що до верхнього шляху. Хронологія: Спочатку ми перейшли у вершину 2, довжина шляху = -4. Потім з вершини 2, ми перейшли у вершину 4, тому що при переході у вершину 5, довжина шляху дорівнюватиме 6, а ми вже знаємо що в нас є довший шлях(8). Отже перебуваючи у вершині 4 маємо галуження: піти у вершину 3 чи 3 вершину 5. Оскільки перехід у вершину 5, дасть нам такий же результат як і на минулому кроці (довжина = 6), то має сенс піти у вершину 3. З вершини 3, перехід тільки у вершину 5. Порахуємо яким буде результат у цьому випадку :

$$\Delta_{15} = | \min \{ -4+A_{12}; -2+A_{13}; -3+A_{14} \} | = | \min \{ -4-1-2-3; -2-3; -3-2-3 \} | =$$

$$= | -4-1-2-4 | = | -10 | = 10.$$

Отже необхідна нам відповідь = 10 (рис. 4.30).



Рисунок 4.30 – Правильний шлях

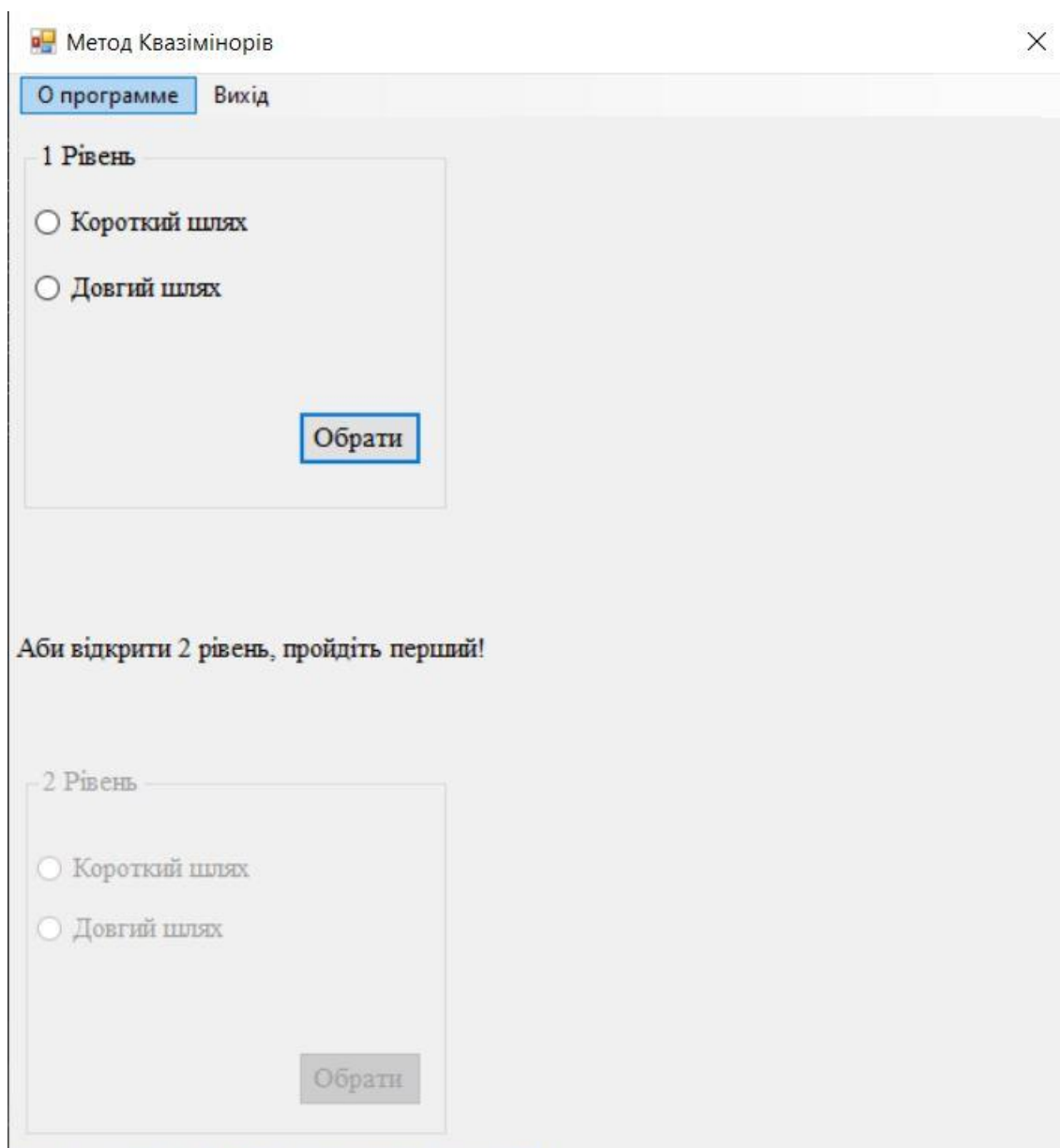


Рисунок 4.31 – Основна сторінка

4.2. Опис програми

Для розробки програми було обрано мову C#, інтерфейс реалізований за допомогою Windows Forms.

Для більшого розуміння роботи програми, надано приклад коду.

Приклад обробки кнопки розпочати:

```
private void button3_Click(object sender, EventArgs e)
```

```

{
    show(5, dataGridView1,0);
    dataGridView1.TopLeftHeaderCell.Value = "Index5";
    dataGridView1.Rows[0].HeaderCell.Value = "1*";
    dataGridView1.Columns[4].HeaderCell.Value = "5*";
    pictureBox1.Visible = true;
    dataGridView1.Visible = true;
    lindex.Visible = true;
    groupBox1.Visible = true;
    button3.Visible = false;
}

```

Далі надано код роботи кнопки вперед:

```

private void button2_Click(object sender, EventArgs e)
{
    switch (cout)
    {
        case 0:
            case0();
            break;
        case 1:
            case1();
            break;
        case 2:
            case2();
            break;
        case 3:
            case3();
            break;
        case 4:
            case4();
            break;
        case 5:
            case5();
            break;
        default:
            break;
    }
}

```

Код було спроектовано таким чином, аби розділити його на короткі та відособлені частини. Реалізація секції switch:

```

private void case0()
{
    if (compare(5, mass, dataGridView1))
    {
        MessageBox.Show("Дуже добре, відповідь вірна, та оскільки це вже другий блок тесту, то ми тришки ускладнемо завдання. :)", "Відповідь вірна", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
        show(5, dataGridView1, 2);
        dataGridView1.Rows[0].HeaderCell.Value = "1*";
        dataGridView1.Columns[4].HeaderCell.Value = "5*";
        show(4, dataGridView2, 3);
        dataGridView2.Rows[1].HeaderCell.Value = "3*";
        dataGridView2.Columns[3].HeaderCell.Value = "5*";
    }
}

```

```

        dataGridView2.TopLeftHeaderCell.Value = "Index4";
        label1.Visible = true;
        dataGridView1.Enabled = false;
        groupBox1.Text = "Знайдіть найкоротший шлях, з вершини 1 у вершину
5.\nПропоную розпочати з вершини 3, оскільки в цьому стовбчику - найменший елемент в
рядочку. Впишіть, яка матриця вийде, якщо перейти з вершини 1 у вершину 3?" +
        "\nЗверніть увагу, що ваги шляхів вже не 1 та 0!!";
        cout++;
        _rating(a);
        a = 2;
        progressBar1.Value = 20;
    }
    else if (a == 0)
    {
        MessageBox.Show("Занадто багато спроб. Матриця яка повинна у вас вийти:" +
            "\n 0,1,1,1,0" +
            "\n 0,0,0,1,1" +
            "\n 0,0,0,0,1" +
            "\n 0,0,1,0,1" +
            "\n 1,0,0,0,0", "Підказка", MessageBoxButtons.OK,
        MessageBoxIcon.Warning);
    }
    else
    {
        MessageBox.Show("Невже це так складно? Як ти сюди потрапив? Кількість спроб
які в тебе ще залишилися: " + a, "Відповідь не вірна!", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
        a--;
    }
}
private void case1()
{
    if (compare(4, mass13, dataGridView2))
    {
        MessageBox.Show("Чудово!\nОтже в нас залишився ще один крок, якою буде
фінальна вага цього шляху, якщо перейти у вершину 5?", "Відповідь вірна",
        MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
        dataGridView2.Enabled = false;
        groupBox1.Text = "Якою буде сума цього шляху?";
        radioButton1.Visible = true;
        radioButton2.Visible = true;
        radioButton3.Visible = true;
        radioButton4.Visible = true;
        cout++;
        _rating(a);
        a = 2;
        progressBar1.Value = 40;
    }
    else if (a == 0)
    {
        MessageBox.Show("Занадто багато спроб. Матриця яка повинна у вас вийти:" +
            "\n 0,0,1,2" +
            "\n 0,0,0,3" +
            "\n 0,0,0,1" +
            "\n 2,0,0,0", "Підказка", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
    else
    {

```



```

        MessageBox.Show("Не розумію, адже ти вже стільки пройшов, я гадав що ти вже
все умієш. Кількість спроб які в тебе ще залишилися: " + a, "Відповідь не вірна!",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        a--;
    }
}
private void case2()
{
    if (radioButton3.Checked)
    {
        MessageBox.Show("Добре, отже довжина цього шляху = 5. Але це не означає що
він найкоротший. Повернемося до першої матриці, та підемо у вершину 4, оскільки елемент
цього стовбчика наступний найменший елемент рядка в якому ми знаходимося.", "Відповідь
вірна", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
        show(4, dataGridView2, 4);
        dataGridView2.Rows[2].HeaderCell.Value = "4*";
        dataGridView2.Columns[3].HeaderCell.Value = "5*";
        dataGridView2.Enabled = true;
        groupBox1.Text = "Яка матриця вийде, якщо перейти з вершини 1 у вершину 4?";
        radioButton1.Visible = false;
        radioButton2.Visible = false;
        radioButton3.Visible = false;
        radioButton4.Visible = false;
        radioButton3.Checked = false;
        label1.Text = "δ15 = 3 +";
        cout++;
        _rating(a);
        a = 2;
        progressBar1.Value = 60;
    }
    else
    {
        MessageBox.Show("Один з елементів знаходиться біля матриці, другий шукай за
допомогою * .", "Помилка!", MessageBoxButtons.OK, MessageBoxIcon.Error);
        a--;
    }
}
private void case3()
{
    if (compare(4, mass14, dataGridView2))
    {
        MessageBox.Show("Матриця заповнена вірно. Отже ми у вершині 4, та бачимо що
маємо 2 доступні шляхи. Оскільки ми в попередньому завданні вже були у вершині 3, то
очевидно що переходячи у цю вершину, " +
        "шлях не буде найкоротшим. Чому буде дорівнювати пройдений шлях, коли ми
потрапимо у вершину 5 з вершини 4?", "Відповідь вірна", MessageBoxButtons.OK,
MessageBoxIcon.Asterisk);
        groupBox1.Text = "Чому буде дорівнювати пройдений шлях, коли ми потрапимо у
вершину 5 з вершини 4?" +
        "\nВідповідь запиши цифрою." +
        "\nПодумайте, чи варто нам іти з вершини 1 у вершину 2.";
        label2.Visible = true;
        textBox1.Visible = true;
        cout++;
        _rating(a);
        a = 2;
        progressBar1.Value = 80;
    }
    else if (a == 0)
    {

```

```

        MessageBox.Show("Занадто багато спроб. Матриця яка повинна у вас вийти:" +
            "\n 0,0,0,2" +
            "\n 0,0,0,3" +
            "\n 0,0,2,1" +
            "\n 2,0,0,0", "Підказка", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
    else
    {
        MessageBox.Show("Помилка? Тут? Це вже 4 завдання, ти що, на приколі?.
Кількість спроб які в тебе ще залишились:" + a, "Відповідь не вірна!", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        a--;
    }
}
private void case4()
{
    if (textBox1.Text == "4")
    {
        MessageBox.Show("Так. Тепер оберіть найкоротший шлях.", "Відповідь вірна",
        MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
        groupBox1.Text = "Оберіть найкоротший із запропонованих шляхів.";
        textBox1.Visible = false;
        label2.Visible = false;
        radioButton1.Visible = true;
        radioButton2.Visible = true;
        radioButton3.Visible = true;
        radioButton4.Visible = true;
        radioButton1.Text = "1-3-5";
        radioButton2.Text = "1-2-5";
        radioButton3.Text = "1-2-4-5";
        radioButton4.Text = "1-4-5";
        cout++;
        _rating(a);
        a = 2;
        progressBar1.Value = 100;
    }
    else
    {
        MessageBox.Show("Маєш значення першого переходу біля матриці, другий шукай у
ній, тобі допоможе *", "Хибна відповідь!", MessageBoxButtons.OK, MessageBoxIcon.Error);
        a--;
    }
}
private void case5()
{
    if (radioButton4.Checked)
    {
        _rating(a);
        MessageBox.Show("Так, це найкоротший шлях." +
            "\nНабрана кількість балів : " + rating, "Результат",
        MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
        groupBox1.Text = "Кількість балів : " + rating;
        radioButton1.Visible = false;
        radioButton2.Visible = false;
        radioButton3.Visible = false;
        radioButton4.Visible = false;
        button2.Enabled = false;
        button1.Visible = true;
        progressBar1.Value = 120;
    }
}

```

```

else
{
    MessageBox.Show("Ми його знайшли на попередньому кроці :)", "Хибна
відповідь!", MessageBoxButtons.OK, MessageBoxIcon.Error);
    a--;
}
}

```

Приклад інших, «не менш важливих» частин коду:

```

private void hide()
{
    pictureBox1.Visible = false;
    dataGridView1.Visible = false;
    lindex.Visible = false;
    groupBox1.Visible = false;
    cout = 0;
}
private void show(short size, DataGridView data, short index)
{
    data.Rows.Clear();
    data.ColumnCount = size;
    data.RowCount = size;
    for (int i = 0; i < size; i++)
    {
        for (int j = 0; j < size; j++)
        {
            data.Columns[j].HeaderCell.Value = Convert.ToString(j + 1);
            data.Rows[i].HeaderCell.Value = Convert.ToString(i + 1);
            data.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;
            data.AutoSizeRowsMode = DataGridViewAutoSizeRowsMode.AllCells;
            if (index == 2)
            {
                data.Rows[i].Cells[j].Value = remass[i, j];
            }
            if (index == 3)
            {
                data.Rows[i].HeaderCell.Value = Convert.ToString(i + 2);
                if (j >= 2)
                {
                    data.Columns[j].HeaderCell.Value = Convert.ToString(j + 2);
                }
            }
            if (index == 4)
            {
                data.Rows[i].HeaderCell.Value = Convert.ToString(i + 2); ;
                if (j == 3)
                {
                    data.Columns[j].HeaderCell.Value = Convert.ToString(j + 2);
                }
            }
        }
    }
    data.Visible = true;
}
private bool compare(short size, int[,] arr2, DataGridView data)
{
    try
    {

```

```

int[,] arr = new int[size, size];
int c = 0;

for (int i = 0; i < size; i++)
{
    for (int j = 0; j < size; j++)
    {
        arr[i, j] = Convert.ToInt32(data.Rows[i].Cells[j].Value);
    }
}
for (int i = 0; i < size; i++)
{
    for (int j = 0; j < size; j++)
    {
        if (arr[i, j] == arr2[i, j] &&
(Convert.ToString(data.Rows[i].Cells[j].Value) != ""))
        {
            c++;
        }
        else c--;
    }
}
if (c == size * size)
{
    return true;
}
else
    return false;
}
catch (Exception)
{
    return false;
}
}

```

ВИСНОВКИ

В бакалаврській роботі було досліджено тему «Метод квазімініорів для відшукування найдовшого та найкоротшого шляхів».

Було здійснено огляд задачі про найкоротший шлях та схожих за призначенням алгоритмів.

Були продемонстровані декілька прикладів з роботи програми.

Було з'ясовано значення терміну квазімініор.

Проведено аналіз недоліків і переваг використання тренажерів.

Створена блок-схема алгоритму.

Написана програма для поглиблення знань студента в предметній області. Програма перевірена, усі помилки усунені.

Було детально проаналізовано роботу програми та створено інструкцію по роботі з тренажером.

Було висвітлено внутрішню оболонку програми, на прикладі блоку коду однієї функціональної особливості.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Математические модели в управлении: учебное пособие / В. П. Заболотский, А.А. Оводенко, А.Г. Степанов – К. : НАУ, 2014. – 13 с.
2. CLR via C#. Программирование на платформе Microsoft.NET Framework 4.5 на языке C#. 4-е изд. / Дж. Рихтер, 2018. – 896 с. – (ISBN: 9785496004336).
3. Л. И. Кича, Е. И. Литвак, “Об одном алгоритме перечисления простых цепей в графе”, *Ж. вычисл. матем. и матем. физ.*, **15**:4 (1975), 1077–1079
4. Комиссаров Ю.А. Анализ и синтез систем водообеспечения химический производств – М.:Химия, 2002. – 496 с. – (ISBN 5-7245-1085-5).
5. Авондо-Бодино Дж. Применение в экономике теории графов. М.: Прогресс, 1966. 160 с..
6. Кирсанов М.Н. Графы в MAPLE. Задачи, алгоритмы, программы. М.: Физматлит, 2007. 91 с.
7. Психология влияния / Р. Чалдини, 2016.-336 с. - (ISBN: 978-5-496-00163-2).
8. Linux глазами хакера. – 4-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2016. – 432 с. - (ISBN: 978-5-9775-3333-1).